



OSSMETER

Automated Measurement and Analysis of Open Source Software

Project Number 318736

D6.4 – Eclipse.org Use Case Evaluation

**Version 1.2
27 July 2015
Final**

Public Distribution

Tecnalia

Project Partners: Centrum Wiskunde & Informatica, SOFTEAM, Tecnalia Research and Innovation, The Open Group, University of L'Aquila, UNINOVA, University of Manchester, University of York, Unparallel Innovation

Every effort has been made to ensure that all statements and information contained herein are accurate, however the Partners accept no liability for any error or omission in the same.

© 2015 Copyright in this document remains vested in the OSSMETER Project Partners.

PROJECT PARTNER CONTACT INFORMATION

<p>Centrum Wiskunde & Informatica Jurgen J. Vinju Science Park 123 1098 XG Amsterdam, Netherlands Tel: +31 20 592 4102 E-mail: jurgen.vinju@cwi.nl</p>	<p>Softeam Alessandra Bagnato Avenue Victor Hugo 21 75016 Paris, France Tel: +33 1 30 12 16 60 E-mail: alessandra.bagnato@softeam.fr</p>
<p>Tecnalia Research and Innovation Jason Mansell Parque Tecnológico de Bizkaia 202 48170 Zamudio, Spain Tel: +34 946 440 400 E-mail: jason.mansell@tecnalia.com</p>	<p>The Open Group Scott Hansen Avenue du Parc de Woluwe 56 1160 Brussels, Belgium Tel: +32 2 675 1136 E-mail: s.hansen@opengroup.org</p>
<p>University of L'Aquila Davide Di Ruscio Piazza Vincenzo Rivera 1 67100 L'Aquila, Italy Tel: +39 0862 433735 E-mail: davide.diruscio@univaq.it</p>	<p>UNINOVA Pedro Maló Campus da FCT/UNL, Monte de Caparica 2829-516 Caparica, Portugal Tel: +351 212 947883 E-mail: pmm@uninova.pt</p>
<p>University of Manchester Sophia Ananiadou Oxford Road Manchester M13 9PL, United Kingdom Tel: +44 161 3063098 E-mail: sophia.ananiadou@manchester.ac.uk</p>	<p>University of York Dimitris Kolovos Deramore Lane York YO10 5GH, United Kingdom Tel: +44 1904 325167 E-mail: dimitris.kolovos@york.ac.uk</p>
<p>Unparallel Innovation Nuno Santana Rua das Lendas Algarvias, Lote 123 8500-794 Portimão, Portugal Tel: +351 282 485052 E-mail: nuno.santana@unparallel.pt</p>	

DOCUMENT CONTROL

Version	Status	Date
0.1	Table of contents and initial descriptive content	16 February 2015
0.9	Evaluation results data	26 March 2015
1.0	Final content integration	5 May 2015
1.1	Additional updates and QA	27 May 2015
1.2	Minor revisions to address final EC review comments	27 July 2015

TABLE OF CONTENTS

1. Eclipse.org Use Case Evaluations	1
1.1 Demonstrator Overview	1
1.2 Evaluation Scenarios	2
1.3 Evaluation Team	3
2. Industrial Validation Measures	4
2.1 Plug-in selection satisfaction	4
2.2 OSS project comparison satisfaction	6
2.3 Service extensibility satisfaction	8
3. Adoption Readiness Measures	10
3.1 Forge extension effort assessment	10
3.2 Communications analysis coverage assessment	11
3.3 Dashboard ease of use satisfaction	12
3.4 Platform performance satisfaction	14
4. Industrial Requirements Fulfilment	15
4.1 General requirements	15
4.2 OSS Project level requirements	16
4.3 Lifecycle related requirements	18
4.4 Code quality requirements	19
4.5 OSS Forum activity requirements	21
4.6 Tracking systems requirements	22
4.7 Platform configurability requirements	22
4.8 Platform extensibility requirements	22
4.9 Overall requirements evaluation	22
5. Future Enhancements	23
6. Conclusions	23

EXECUTIVE SUMMARY

This report documents the evaluation results of the improvements provided by OSSMETER technologies based on deployment of the project technologies in undertaking industrial grade use cases addressing requirements in the selection, evaluation and monitoring of open source software projects, and the measurement of the level of achievement with respect to the target measures identified in deliverable *D6.1 – Specification of Use Case and WP evaluation criteria*.

The specific Use Case for this use case is described and the evaluation procedures undertaken and quantitative results are provided in this report. In general the evaluation of the OSSMETER technologies showed the highest priority industrial requirements for this use case as described in deliverable *D1.1 – Project Requirements*, have been largely addressed and that amongst the industrial evaluation measures the project technologies achieved an Excellent score in five out of the seven metrics for this use case.

The report also identifies some additional features such as OSSMETER technologies applied onto existing Tecnalía's services, which were not part of the project work plan and can be easily included during the exploitation phase following completion of the project.

1. ECLIPSE.ORG USE CASE EVALUATIONS

1.1 DEMONSTRATOR OVERVIEW

TECNALIA is the first applied research centre in Spain and one of the most important in Europe with around 1.500 staff, 122 million Euro turnover and over 4.000 clients. Inside Tecnalía, the ICT/ESI Competitiveness division develops software to increase the productivity of its customers. This area focuses its activity on:

- Service companies, to make a smart use of ICTs that provide unique experiences to our users.
- Companies that produce products and solutions of all sorts, so that the ICTs can place outstanding solutions and products in the market.
- IT companies, to efficiently produce the best software systems, that are more robust, reliable, and better managed and more user friendly.
- Other sectors, to exploit new ICT-based concepts that radically replace traditional processes or products.

In order to fulfil these objectives, Tecnalía has based many of its developments on Eclipse technologies. Some of these projects are as follows:

- GEMDE: Generic executable model-driven engineering framework
- PLUM: Product Lines generator
- FAST: Factory of Software
- ProcessFactory: Business process generator and instantiator
- SMOOL: Smart spaces creation platform

Tecnalía is interested in Free/Open Source Software in several ways:

- **Measuring Eclipse ecosystem:** Tecnalía ICT/ESI Competitiveness division develops products and plugins based on Eclipse software and has a special interest in measuring several Eclipse projects to make strategic decisions about using them or integrating them in our own projects.
- **Software development:** in software development there is no need to reinvent the wheel and many times there is an OSS project (or many) that implement a functionality that is needed. OSSMETER is expected to assist in taking such decisions by providing comprehensive metrics.
- **Community impact and Quality assurance:** Tecnalía plans to use the information extracted from the OSSMETER project to measure the real impact of its own OSS public developments. OSSMETER will also help to ensure the quality of this software.
- **Certification:** Tecnalía thinks that there can be a business opportunity in Software Certification related to the metrics extracted from OSSMETER analysis. Specific metric profiles and thresholds could be specified for compliance and certification purposes.

- **Internal use:** there are many servers, services and programs used in Tecnalia that are FLOSS, so OSSMETER will help Tecnalia to choose between different OSS alternatives for its own use.

1.2 EVALUATION SCENARIOS

Tecnalia implemented two different evaluation scenarios: one to select an Eclipse-plugin for a new project and another one to present a comparative report of one of Tecnalia’s Eclipse plugins to a client.

Scenario 1: OSSMETER dashboard helps in selecting OSS software alternatives for a new project

In the first scenario, two different groups of the Tecnalia ICT-ESI division had to choose an Eclipse plugin to be used in a hypothetical MDE project, providing a rationale about the decision taken. One group used the desired OSSMETER workflow for the selection and the other one used the current workflow. Each group worked independently and did not know about the other one’s work.

After the selection, both rationales were compared and both groups decided together which one was the best solution.

OSSMETER enriched the decision and the group working with the OSSMETER workflow reached a better, or at least a more substantiated decision.

Scenario 1: Selecting Eclipse plugin alternatives	
Actors	Group-1 , Group-2 (at least one manager and two developers per group)
Current Workflow	<ol style="list-style-type: none"> 1) Group-1 studies the tools/OSS components/ plugins required for the specific project 2) Group-1 looks for previously used plugins in the same context 3) Group-1 looks OSS software alternatives browsing engines on the internet. 4) All the OSS elements retrieved are analysed and tested.
OSSMETER Workflow	The different OSS component would be selected by Group-2 evaluating the metrics provided by OSSMETER.

Table 1 – Scenario 1: Selecting Eclipse plugin alternatives for new project

Scenario 2: OSSMETER dashboard helps in enriching existing plugin information in a commercial offer

For the second scenario, Tecnalia simulated a commercial offer for one of Tecnalia’s Eclipse plugins and presented it to a current client. For the first version of the offer the current plugin information and commercial information was used. The second version was enriched with OSSMETER information from as many metrics as possible. The client completed a questionnaire / evaluation about both versions of the offer.

OSSMETER enriched the plugin offer and provided additional useful information. The main objective of the exercise was to measure the satisfaction of the client about the

additional information and evaluate if this additional information could make a difference between purchasing, or not, a Tecnalía product.

The specific plugin was FAST (Factory of Software) Eclipse plugin and the client was a current Tecnalía client (Ibermatica).

Scenario 2: Enrich plugin information in an offer	
Actors	Developers, Manager, Consultant
Current Workflow	Manager and Developers prepare a complete commercial offer (version one) to simulate a contract with a client with specific needs.
OSSMETER Workflow	After the evaluation of the offer (version one) the manager and the Developers enrich the offer with OSSMETER dashboard information and send this version (version two) to the client to re-evaluate it. A Consultant evaluates the client's satisfaction with both offers and generates an evaluation report about the value of the OSSMETER information provided in the version two.

Table 2 – Scenario 2: Eclipse plugin information enrichment

1.3 EVALUATION TEAM

The evaluation team / group for OSSMETER to all the scenarios was composed by the same people with different skills. The group includes people involved in OSSMETER and developers not in the project.

	Role	Relevant skills	OSSMETER participation
Jason Mansell (JM)	Project Manager, Lead investigator	Junior Java and Eclipse plug-in developer	yes
Alberto Berreteaga (AB)	Project Manager, Investigator, Engineer	Senior Java and Eclipse plug-in developer	yes
Guillermo Rodriguez (GR)	Investigator, Engineer	Senior Java and Eclipse plug-in developer	yes
Angel Rego (AR)	Investigator, Engineer	Senior Java developer and Junior Eclipse plug-in developer	yes
Eider Iturbe (EI)	Engineer	Senior Java and Eclipse plug-in developer	no

Gorka Benguria (GB)	Investigator, Engineer	Senior Java and Eclipse plug-in developer	no
Gorka Mikel Echevarria (GE)	Engineer	Senior Java developer and Junior Eclipse plug- in developer	no
Juan Bartolomé (JB)	Engineer	Junior Java and Eclipse plug-in developer	no

2. INDUSTRIAL VALIDATION MEASURES

The measures for industrial validation identify the degree to which the project has implemented expected technology innovations for evaluating open source software projects publicly available online, or available in-house for a software development organisation. These measures quantify the level of improvements achieved and therefore the expected industry impact the OSSMETER technologies should have for organisations interested in or already using open source software. In addition, they indicate the degree to which the project development tasks have delivered technology innovations that address the needs of typical industrial organisations in this domain.

2.1 PLUG-IN SELECTION SATISFACTION

Justification

An important objective for the OSSMETER technologies is to provide a more robust set of metrics on which the selection of Eclipse plug-ins for a particular use or application scenario can be based. The dashboard is the main interface where users interact with the OSSMETER technologies so it's important that the interface is user friendly and provides an effective environment in which to analyse the Eclipse plugins and projects. The dashboard should be perceived by the users as being easy to use and easily customisable.

Specification

A group of Tecnalia ICT-ESI Division developers who are already experienced in selecting Eclipse plug-ins in previous Tecnalia projects have been given a scenario (intended application, main features, etc.) where they each need to select an Eclipse plug-in to be used in a hypothetical new MDE project. The scenario was designed such that several potential Eclipse plug-ins could be appropriate in terms of satisfying the desired functionalities. The developers utilised the OSSMETER technologies to carry out a decision making process to select the Eclipse plug-in that is deemed best suited for the hypothetical scenario.

After the plug-in selection was completed, a survey instrument was used to collect the views of the developers relative to their existing procedures for selecting Eclipse plug-ins and included the following questions:

Compared to previous plug-in selection processes within Tecnalía,

- Q1 : How would you rate the quantity of information available to support your decision making?
- Q2 : How would you rate the quality of the information available to support your decision making?
- Q3 : How would you rate the effort required to make your decision?
- Q4 : How would you rate the ability to tailor the information to support your decision making?
- Q5 : How would you rate the ability to define custom criteria for your decision making?
- Q6 : How would you rate your overall confidence in the decision for the selected plug-in?

In addition, there was an opportunity for developers to provide comments and clarifications concerning their experiences and justify their ratings in each category.

The scale for answering the questions was the following:

- High: there was a large improvement
- Medium: there was a moderate improvement
- Low: there was a small improvement
- None: there was little or no improvement

The following evaluation scale specified in deliverable *D6.1 – Evaluation Measures* has been applied to determine and overall rating of the measure for this use case:

Questionnaire responses	Evaluation
All responses high	Excellent
All responses medium or high	Good
Majority of responses medium or high	Sufficient
Majority of responses low or none	Insufficient

The following comments were received when collecting the survey results:

- It was difficult to define the “similarity” between the plug-ins. Different plug-ins had different features so a more detailed initial MDE project definition was really important.
- Previous individual knowledge of the plug-ins was important. Even if the metrics of one unknown plug-in were better than those of a known one the confidence about the known one was higher. This “fear of the unknown” factor was more important than expected.

	Q1	Q2	Q3	Q4	Q5	Q6
--	----	----	----	----	----	----

JM	excellent	good	good	good	good	excellent
AB	excellent	good	good	good	good	excellent
GR	excellent	excellent	excellent	good	good	good
AR	good	good	good	good	excellent	good
EI	good	good	good	good	good	good
GB	good	good	good	good	good	good
GE	excellent	good	good	good	good	good
JB	good	good	good	good	good	good

Overall the project technologies scored a rating of Good.

2.2 OSS PROJECT COMPARISON SATISFACTION

Justification

A key feature of the OSSMETER platform required for industrial use is the ability to select and compare multiple OSS projects using a variety of analysis metrics. This allows users of OSS to make more informed choices and to customise selection criteria to be applied to each of the potential OSS project candidates. This is especially beneficial for OSS projects that might be hosted in different forges, making it difficult to easily apply common criteria for use in comparison and selection of OSS projects.

Specification

Using the same procedure specified in Section 2.1 where Tecnalia developers have been given a scenario (intended application, main features, etc.) where they each need to select an Eclipse plug-in to be used in a hypothetical new MDE project, a survey instrument was used to collect the views of the developers relative to their existing procedures for selecting Eclipse plug-ins and included the following questions.

Compared to previous plug-in comparison procedures within Tecnalia,

- Q1 : How would you rate the quantity of plug-ins you were able to compare simultaneously?
- Q2 : How would you rate the quantity of criteria you used for comparing plug-ins?
- Q3 : How would you rate the speed at which you were able to carry out the comparison of plug-ins?
- Q4 : How would you rate the ability to customise the comparison of plug-ins?

In addition, there was an opportunity for Use Case developers to provide comments and clarifications concerning their experiences and why they chose specific ratings in each category.

The scale for answering the questions was the following:

- High: there was a large increase
- Medium: there was a moderate increase
- Low: there was a small increase
- None: there was little or no increase

The following evaluation scale specified in deliverable *D6.1 – Evaluation Measures* has been applied to determine and overall rating of the measure for this use case:

Questionnaire responses	Evaluation
All responses high	Excellent
All responses medium or high	Good
Majority of responses medium or high	Sufficient
Majority of responses low or none	Insufficient

	Q1	Q2	Q3	Q4
JM	excellent	excellent	excellent	excellent
AB	excellent	excellent	excellent	excellent
GR	excellent	excellent	excellent	good
AR	excellent	good	good	excellent
EI	excellent	excellent	good	good
GB	excellent	good	excellent	good
GE	excellent	excellent	good	excellent
JB	excellent	excellent	good	excellent

Overall the project technologies scored a rating of Excellent.

The team worked with different metrics and Eclipse projects (Maven, BPEL Designer, EMF, Hudson) but to avoid comparisons between “different” plug-ins, plug-ins with different features or functions, the team focus the evaluation onto some different versions of the Eclipse java IDE:

- Luna

- Kepler
- Juno
- Indigo

The metrics over the different versions were the same ones and the most relevant ones were:

- % Bugs solved
- Average time of bug resolution
- Average positive / negative ratio of the comments on the forums

2.3 SERVICE EXTENSIBILITY SATISFACTION

Justification

The OSSMETER project is providing an integrated platform for evaluating OSS projects using a variety metrics. However, the platform is based on an open architecture that is intended to allow OSSMETER technologies and analysis metrics to be applied in many different scenarios. In particular, the capability of OSSMETER to provide a useful metrics model that can be used without the OSSMETER dashboard opens opportunities to customised analytics and other services that exploit the OSSMETER technologies in application scenarios beyond those initially targeted by the project. The openness and the ability to exploit the underlying analysis technologies is therefore an important measure

Specification

A group of Tecnalía ICT-ESI Division developers was asked to develop a prototype analysis tool that interfaces with the OSSMETER platform to utilise a small subset of the available analysis metrics. The prototype was only a proof of concept so that developers could verify the REST-API interface functionalities without investing significant effort in secondary areas such as the appearance of the prototype tool. The metrics used included at least one metric from each of the code analysis and communication channel metrics.

The developed application provides to a consultancy group about when new people enter the project and when people leaves (or at least provides no more code to the repository). The idea is to check the people rotation during a project lifetime and their mood during different time slots.

The main metrics were related to:

- % of activity of a user during a project (code upload / time)
- Average people rotation time (when a new developer is added, when a developer leaves the project, etc.)

- Average positive/negative comments of a developer during the its activity (in slots of 25% of the complete time)

In addition to the people rotation data extraction, the idea is try to evaluate if a “new” developer is positive/negative during its first period (24% of the time) on the project and how its mood evolves.

After completing the development of the prototype analysis tool, a survey instrument was used to collect the views of the developers with regard to the OSSMETER platform and REST API provided with the following questions:

- Q1 : The platform REST API sufficiently specified to implement the prototype analysis tool.
- Q2 : I was able to access the code analysis metrics that were required.
- Q3 : I was able to access the communications channel metrics that were required.
- Q4 : The response times for the prototype analysis tool were sufficient using the REST-API.
- Q5 : The data provided by the platform didn’t require additional processing before it could be utilised by the prototype analysis tool.

In addition, there was an opportunity for developers to provide comments and clarifications concerning their experiences and why they chose specific ratings in each category.

The scale for answering the questions was the following:

- Fully agree
- Largely agree
- Partially agree
- Don’t agree

The following evaluation scale specified in deliverable *D6.1 – Evaluation Measures* has been applied to determine and overall rating of the measure for this use case:

Questionnaire responses	Evaluation
All responses fully positive	Excellent
All responses positive	Good
Majority of responses positive	Sufficient
Majority of responses negative	Insufficient

The following comments were received when collecting the survey results:

- More options about partial / current processing status of the information needed. Even if the platform is able to launch alarms when it completes the analysis some developers ask for partial / current status information or completion estimations.

	Q1	Q2	Q3	Q4	Q5
JM	excellent	excellent	excellent	good	excellent
AB	excellent	excellent	excellent	excellent	excellent
GR	excellent	excellent	excellent	good	excellent
AR	excellent	good	excellent	good	good
EI	excellent	excellent	good	good	good
GB	excellent	excellent	excellent	good	good
GE	good	excellent	good	excellent	excellent
JB	excellent	excellent	good	excellent	excellent

Overall the project technologies scored a rating of Excellent.

3. ADOPTION READINESS MEASURES

The OSSMETER technologies are intended to be exploited in a variety of scenarios ranging from free or commercial online OSS assessment services, to in-house monitoring of software quality. Key technology capabilities developed in the research and development workpackages have been evaluated to ensure the project results will address the needs of industry, meet expected levels of performance and usability, and that the project results can be tailored to address a wide range of industrial scenarios and a diverse set of OSS projects and forges.

3.1 FORGE EXTENSION EFFORT ASSESSMENT

Justification

The OSSMETER project has targeted many of the most popular OSS forges used by industry today for carrying out evaluations using the new project technologies. However, the partners recognise that there are many more forges being used to host OSS projects ranging from generic and public forges that have no specific focus to others targeted at specific industrial sectors or application domains. A key factor in the take-up of the OSSMETER technologies by industry is the ability to interoperate with and provide evaluations for OSS projects hosted by forges beyond those specifically targeted by the project. Assurances are needed for public or commercial service providers using OSSMETER technologies to provide OSS evaluations, as well as in-house developers that the OSS forges where they have specific interests in evaluating OSS projects can be accommodated by the OSSMETER platform.

Specification

A small team of Tecnia developers was assigned to extend the OSSMETER platform to support analysis of OSS projects from an additional forge (Sourceforge and inner

TecNALIA customized Git structure). The effort required to implement the functioning extension of the platform to an additional forge was monitored. The selected additional forge was similar to the forges already supported so that the focus was on extension and minor development tasks, rather than new research and development for support of substantially different forges beyond the scope of the project.

After completing the extension for an additional forge the total time required was calculated and the following evaluation scale specified in deliverable *D6.1 – Evaluation Measures* has been applied to determine an overall rating of the measure for this use case:

Forge Extension Effort	Evaluation
Less than or equal 5 person days	Excellent
Between 5 and 10 person days	Good
Between 10 and 20 person days	Sufficient
More than 20 person days	Insufficient

The component development final effort depends not only on the OSSMETER technologies but on each person's skills and knowledge about Java and Eclipse plug-ins too. The OSSMETER documentation to extend the forges is to be good and enough for all the developers, even for the junior ones.

	Effort
JM	12 p/d
AB	5 p/d
GR	2 p/d
AR	2 p/d
EI	7 p/d
GB	7 p/d
GE	6 p/d
JB	10 p/d

Overall the project technologies scored a rating of Good.

3.2 COMMUNICATIONS ANALYSIS COVERAGE ASSESSMENT

Justification

The OSSMETER project has invested substantial effort in establishing a baseline for potential communication channel sentiments, events, and other characterisations that

could be relevant in the evaluation of OSS projects. It's important to determine if this baseline is sufficient for carrying out typical OSS project evaluations or whether additional sentiments or situations need to be detected in order to support the types of analysis or the specific areas of interest that industrial organisations would have when evaluating the communications related to a particular set of OSS projects.

Specification

This measure is based on monitoring carried out during the entire validation process of the OSSMETER technologies for this use case. Any event in which the desired sentiment or exchange of interest cannot be addressed by the communication channel analysis tools has been tracked. The change request and tracking system used during the evaluations has been used to enter a change request so that at the completion of the evaluation it was possible to identify the total number of events.

The change request system (<https://github.com/ossmeter/ossmeter/issues>) was used to allow the development partners to review and verify the need for additional channel communications capabilities, or to provide guidance to industrial partners when features were already available but not being properly utilised during the evaluations.

Testing of general communication channel sentiments and events has been performed as described at section 2.2.

The communications analysis coverage measure was calculated, as specified in deliverable *D6.1 – Evaluation Measures*, as the total number of events where additional sentiment or exchange features were needed. It was evaluated using the following scale:

Communications Analysis Extensions	Evaluation
Less than or equal to 5 events	Excellent
Between 5 and 10 events	Good
Between 10 and 20 events	Sufficient
More than 20 events	Insufficient

No additional channel capabilities or extensions were detected, the OSSMETER issues tracker and the Eclipse.org forums and wikis were satisfactory, and so the overall the project technologies scored a rating of excellent.

3.3 DASHBOARD EASE OF USE SATISFACTION

Justification

The OSSMETER tools are intended to eventually be used for industrial OSS analysis and their take-up by industry will depend in part on how easy they are to use and how stable they are in providing the expected features and functions. Developers' expectations for user interface, support for various forges, and configurability have steadily increased and usability must therefore be considered as a relative measure that depends on the existing tool environment and development practices of an organisation.

Specification

A survey instrument was used to collect views of developers as to the usability of the OSSMETER platform and analysis tools. The survey included the following questions:

- Q1 : It was easy to use the platform.
- Q2 : The platform was stable and didn't crash.
- Q3 : It was easy to identify available analysis metrics.
- Q4 : It was easy to understand the analysis results.
- Q5 : It was easy to create a custom analysis profile involving multiple metrics.

In addition, there was an opportunity for developers to provide comments and clarifications concerning their experiences and why they chose specific ratings in each category.

The scale for answering the questions was the following:

- Fully agree
- Largely agree
- Partially agree
- Don't agree

The following evaluation scale specified in deliverable *D6.1 – Evaluation Measures* has been applied to determine and overall rating of the measure for this use case:

Questionnaire responses	Evaluation
All responses fully positive	Excellent
All responses positive	Good
Majority of responses positive	Sufficient
Majority of responses negative	Insufficient

The following comments were received when collecting the survey results:

- As we focus on Eclipse data the initial expectation from the developers was to find Eclipse look-and-feel and Eclipse terminology (although this was not a problem at all for the evaluation).

	Q1	Q2	Q3	Q4	Q5
JM	excellent	excellent	excellent	excellent	excellent
AB	excellent	excellent	excellent	excellent	excellent
GR	excellent	excellent	excellent	good	good
AR	excellent	excellent	excellent	excellent	excellent
EI	excellent	excellent	excellent	excellent	excellent
GB	excellent	excellent	excellent	good	excellent

GE	excellent	excellent	excellent	excellent	excellent
JB	excellent	excellent	excellent	excellent	excellent

Overall the project technologies scored a rating of Excellent.

3.4 PLATFORM PERFORMANCE SATISFACTION

Justification

The OSSMETER platform hosts a wide range of analysis capabilities and utilises background and incremental tasks that periodically analyse and collect metric data from various OSS forges to provide an interactive environment for the analysis of OSS projects. An important element in the adoption of the OSSMETER technologies is the overall impression of the performance of the platform in carrying out OSS project evaluations.

Specification

A survey instrument was used where their views of the developers as to the performance of the OSSMETER platform and analysis tools were collected. The survey included the following questions:

- Q1 : The dashboard met your expectations in terms of responsiveness.
- Q2 : The speed at which single analysis metrics were provided met your expectations.
- Q3 : The time required to carry out an individual OSS project assessment was acceptable.
- Q4 : The time to customise an analysis of an OSS project was acceptable.
- Q5 : There were no steps or tasks to carry out and OSS project evaluation that were too time consuming.

In addition, there was an opportunity for Use Case developers to provide comments and clarifications concerning their experiences and why they chose specific ratings in each category.

The scale for answering the questions will be the following:

- Fully agree
- Largely agree
- Partially agree
- Don't agree

The following evaluation scale specified in deliverable *D6.1 – Evaluation Measures* has been applied to determine and overall rating of the measure for this use case:

Questionnaire responses	Evaluation
All responses fully positive	Excellent

All responses positive	Good
Majority of responses positive	Sufficient
Majority of responses negative	Insufficient

	Q1	Q2	Q3	Q4	Q5
JM	excellent	excellent	excellent	excellent	excellent
AB	excellent	excellent	excellent	excellent	good
GR	excellent	excellent	good	good	good
AR	excellent	excellent	excellent	excellent	excellent
EI	excellent	excellent	excellent	excellent	excellent
GB	excellent	excellent	excellent	excellent	excellent
GE	excellent	excellent	excellent	excellent	excellent
JB	excellent	good	excellent	good	excellent

Overall the project technologies scored a rating of Excellent.

4. INDUSTRIAL REQUIREMENTS FULFILMENT

The OSSMETER framework was driven by industrial user requirements representative of this use case (see deliverable *DI.1 – Requirements Specification*). Priorities of requirements (e.g. Shall, Should, May) have also been specified. While not all requirements may be within the scope of the project, the percentage of requirements that have been satisfied provides an important measure of the applicability of the project results in solving the industrial challenges targeted by the project.

In the following sections each of the industrial requirements for this use case has been evaluated in terms of the extent to which the requirement has been fulfilled. Explanations are provided for requirements that have been noted as partially fulfilled or not fulfilled.

4.1 GENERAL REQUIREMENTS

ID	Requirement	Eclipse Priority	Evaluation	Comment
1	Provide a dashboard that makes possible an overall comparison of OSS projects.	SHALL	Fulfilled	
2	Provide a dashboard that consolidates and displays a	SHOULD	Fulfilled	

ID	Requirement	Eclipse Priority	Evaluation	Comment
	detailed analysis of metrics for each OSS project			
3	Provide documentation describing the capabilities and how to use the OSSMETER service	SHALL	Partially Fulfilled	More documentation and examples for first time users
4	Provide a user interface to OSSMETER services that uses a web browser	SHALL	Fulfilled	
5	Provide a dashboard that is user friendly	SHALL	Fulfilled	
6	Provide a capability for easy navigation through different OSS project metrics	SHALL	Fulfilled	
7	Provide the capability to establish specific thresholds for reporting or displaying selected metrics	SHALL	Fulfilled	
8	Provide notification in case a specific threshold is exceeded.	MAY	Fulfilled	
9	Provide the capability to choose the metrics to be displayed.	SHALL	Fulfilled	

4.2 OSS PROJECT LEVEL REQUIREMENTS

ID	Requirement	Eclipse Priority	Evaluation	Comment
10	Identify the components of the OSS project	SHALL	Not Fulfilled	An OSS project could depend on other OSS projects. This kind of relation is not modelled under the OSSMETER platform.
11	Identify the external "libraries" of the OSS project	SHALL	Not Fulfilled	
12	Provide the ability to monitor only the activity of selected OSS project components	MAY	Partially Fulfilled	Each project component has to be an OSS project. The aggregation of these components into one top components is not

ID	Requirement	Eclipse Priority	Evaluation	Comment
				possible
13	Provide an indicator of the age of the OSS project components	SHOULD	Partially	Overall age of the OSS and versions can be determined.
14	Provide a count of the number of downloads of the OSS project	MAY	Fulfilled	If the OSS project has not this information or cannot be calculated it will not be displayed
15	Identify the criticality of the OSS project	SHALL	Partially Fulfilled	This metric is not provided by default by the OSSMETER platform but it can be seen as a combination of other metrics or a new metrics implemented by using the infrastructure provided by OSSMETER.
16	Provide a measure of the contributors' commitment to OSS	SHALL	Fulfilled	
17	Provide an overall indicator of the level of OSS project activity	SHALL	Fulfilled	
18	Provide an indicator of the estimated OSS project effort	SHOULD	Fulfilled	
19	Provide an indicator of the number of installations	SHALL	Partially Fulfilled	This metric needs information, which is currently not provided for Eclipse projects.
20	Identify the main contributor/sponsor of the OSS project	SHALL	Fulfilled	
21	Provide an indicator that would represent the enterprise readiness for a an OSS project	SHOULD	Partially Fulfilled	Enterprise readiness does not have a standard definition and depends on end user knowledge. OSSMETER provides an infrastructure

ID	Requirement	Eclipse Priority	Evaluation	Comment
				allowing each end user to implement their specific interpretation of the OSS project enterprise readiness.
22	Provide the Functionalities metric for a an OSS project	MAY	Not Fulfilled	As for #21 this metric does not come with a standard definition and a dedicated implementation but OSSMETER platform provides the environment which allows its implementation by the end user.
23	Provide the capability to set up an alert concerning specific values of a metric for an OSS project	MAY	Fulfilled	
24	Provide the capability to be notified in case thresholds are reached for a metric where an alert has been set up for an OSS project	MAY	Fulfilled	
25	Provide a count of the number of active contributors to the OSS project	SHALL	Fulfilled	
26	Provide a count of the number of releases of the OSS project	SHALL	Fulfilled	
27	Provide a count of the number of active committers to the OSS project	SHALL	Fulfilled	

4.3 LIFECYCLE RELATED REQUIREMENTS

ID	Requirement	Eclipse Priority	Evaluation	Comment
28	Identify the periodicity of the major releases of the OSS project	SHALL	Fulfilled	
29	Indicate the evolution of the community of committers/contributors to the	SHALL	Fulfilled	

ID	Requirement	Eclipse Priority	Evaluation	Comment
	OSS project over time			
30	Provide the ability to identify the response of the OSS project to external events	MAY	Fulfilled	
31	Provide the ability to identify the response of the OSS project community to external events	MAY	Fulfilled	
32	Indicate the evolution of the OSS project over time	SHOULD	Fulfilled	
33	Identify the lifecycle state for OSS projects from forge environments having defined states	SHALL	Fulfilled	

4.4 CODE QUALITY REQUIREMENTS

ID	Requirement	Eclipse Priority	Evaluation	Comment
34	Provide a rating of the quality of code comments of the OSS project	SHALL	Fulfilled	
35	Provide a well-structured code index for the OSS project	SHALL	Fulfilled	
36	Provide a rating of the use of advanced language features for the OSS project	MAY	Fulfilled	
37	Provide a rating of the use of testing cases for the OSS project	SHALL	Fulfilled	
38	Provide an indicator of the possible bugs from empty try/catch/finally/switch blocks for the OSS project	SHALL	Fulfilled	
39	Provide an indicator of the dead code from unused local variables, parameters and private methods for the OSS project	SHALL	Not Fulfilled	This metric is not currently implemented or calculated from existing Eclipse API data.
40	Provide an indicator of the empty if/while statements for the OSS project	SHALL	Fulfilled	

ID	Requirement	Eclipse Priority	Evaluation	Comment
41	Provide an indicator of overcomplicated expressions from unnecessary if statements and for loops that could be while loops for the OSS project	SHALL	Partially Fulfilled	A cyclomatic complexity metric is provided, which is similar in function.
42	Provide an indicator of suboptimal code from wasteful String/StringBuffer usage for the OSS project	SHALL	Not Fulfilled	This metric is not currently implemented or calculated from existing Eclipse API data.
43	Provide an indicator of duplicate code by detecting copied/pasted code for the OSS project	SHALL	Fulfilled	
44	Provide an indicator of the use of Javadoc comments for classes, attributes and methods for the OSS project	SHALL	Fulfilled	
45	Provide an indicator of the use of the naming conventions of attributes and methods for the OSS project	SHALL	Fulfilled	
46	Provide an indicator of the limit of the number of function parameters and line lengths for the OSS project	SHALL	Fulfilled	
47	Provide an indicator of the presence of mandatory headers for the OSS project	SHALL	Fulfilled	
48	Provide an indicator of the use of packets imports, of classes, of scope modifiers and of instructions blocks for the OSS project	SHALL	Fulfilled	
49	Provide an indicator of the spaces between some characters for the OSS project	SHALL	Fulfilled	
50	Provide an indicator of the use of good practices of class construction for the OSS project	SHALL	Fulfilled	
51	Provide an indicator of the use of multiple complexity measurements, among which expressions for the OSS project	SHALL	Fulfilled	
52	Provide an indicator of the	SHALL	Fulfilled	

ID	Requirement	Eclipse Priority	Evaluation	Comment
	cyclomatic complexity for the OSS project			

4.5 OSS FORUM ACTIVITY REQUIREMENTS

ID	Requirement	Eclipse Priority	Evaluation	Comment
53	Provide a set of measures indicating the time between a forum post and subsequent discussions	SHOULD	Fulfilled	
54	Identify the level of activity amongst categories of discussion topics (e.g. support, development, new features, etc.)	SHALL	Fulfilled	
55	Provide an indicator of the overall sentiment within the OSS project community	SHOULD	Fulfilled	
56	Provide an indicator of the sentiment within the OSS project community regarding specific categories of discussion topics	SHALL	Fulfilled	
57	Provide an indicator of the number of registered users amongst the OSS project community	SHALL	Fulfilled	
58	Provide an indicator of the number of active users amongst the OSS project community	SHOULD	Fulfilled	
59	Provide an indicator of the number of community feedback ratings within the OSS project community	SHALL	Fulfilled	
60	Provide an indicator of the continuity of OSS project activities	SHALL	Fulfilled	
61	Provide an indicator of the OSS project community liveliness	SHOULD	Fulfilled	

4.6 TRACKING SYSTEMS REQUIREMENTS

ID	Requirement	Eclipse Priority	Evaluation	Comment
62	Provide a count of the number of issues/bugs reported, open, solved	SHALL	Fulfilled	
63	Provide an indicator of the time between issues/bugs being reported and their being solved or closed	SHALL	Fulfilled	

4.7 PLATFORM CONFIGURABILITY REQUIREMENTS

ID	Requirement	Eclipse Priority	Evaluation	Comment
64	Provide the capability to configure the metric parameters utilised for analyses	SHALL	Partially Fulfilled	This can only be done via a dedicated enhancement of the OSSMETER platform.
65	Provide the capability to install the OSSMETER platform as a standalone system	SHALL	Fulfilled	

4.8 PLATFORM EXTENSIBILITY REQUIREMENTS

ID	Requirement	Eclipse Priority	Evaluation	Comment
66	Provide the capability to extend or customise the dashboard	MAY	Fulfilled	
67	Provide an open API to make available OSSMETER services to other systems	MAY	Fulfilled	
68	Provide the ability to integrate new metric analysers within the OSSMETER platform	MAY	Fulfilled	

4.9 OVERALL REQUIREMENTS EVALUATION

The following evaluation scale specified in deliverable *D6.1 – Evaluation Measures* has been applied to determine an overall rating of the fulfilment of the industrial requirements for this use case:

“Shall” Requirements Implemented	Evaluation
Greater than or equal to 90%	Excellent
Between 80% and 90%	Good

“Shall” Requirements Implemented	Evaluation
Between 70% and 80%	Sufficient
Less than 70%	Insufficient

The OSSMETER technologies have achieved an overall pondered rating of Excellent in fulfilling the industrial requirements for this use case.

5. FUTURE ENHANCEMENTS

A secondary objective of the evaluation process was to identify any improvements that can be used to provide additional benefits to users in this use case. During the evaluations some minor improvements were identified that can contribute to the evolution of the OSSMETER technologies, which were not within the original scope of the project development work plan. These include the following:

- Manage related projects, or dependencies between projects, to generate average or “composited” metrics information.
- More social networks feature in the user profile. Features to share info, activities, metrics, etc.
- Similar / same metrics identification and statistics.
 - If somebody has defined / uses the same metric other user has previously defined show some statistical information about how many other people uses the metric, over which projects and similar information.
 - Browse existing metrics in different projects using key words / metadata to get ideas about new or variation metrics.
 - If a metric already exists / has been previously calculated show “last calculated value” before recalculate it for the user.
 - ~~○ metrics to Share / look for existing metrics to avoid duplicate calculations and / or get ideas about metrics that other people are using and that can be useful for other user.~~

These enhancements do not diminish the expected impact of the project technologies amongst developers who are considering or already rely on OSS, but could provide additional benefits in coming versions of the OSSMETER platform.

6. CONCLUSIONS

In the industrial validation Tecnalia’s measures quantify the level of improvements achieved and therefore the expected industry impact the OSSMETER technologies. With two Excellent evaluations and one Good evaluation out of three assessments, OSSMETER technologies seem to successfully address the needs expressed through these evaluations.

On the three assessments (Plug-in selection satisfaction, OSS project comparison satisfaction and Service extensibility satisfaction) a group of Tecnalía's developers worked together using OSSMETER technologies and completed a survey to evaluate the satisfaction levels.

In the adoption readiness Tecnalía measures if the project results will address the needs of industry, meet expected levels of performance and usability, and the extent to which the project results can be tailored to address a wide range of industrial scenarios and a diverse set of OSS projects and forges. With two Excellent evaluations and two Good evaluations out of four assessments, OSSMETER technologies seem to successfully address the needs expressed through these evaluations.

A group of Tecnalía developers have performed two assessments (Forge extension effort assessment and Communications analysis coverage assessment) and have completed two surveys about dashboard ease of use satisfaction and platform performance satisfaction.

The Industrial Requirements Fulfilment section presents the fulfilment evaluation of the industrial user requirements representative of the Eclipse.org use case. OSSMETER industrial users defined 68 requirements and almost all of them result in fulfilment, which corresponds to an overall rating of Excellent. Furthermore, the partially fulfilled and not fulfilled requirements are not critical or particularly relevant in the OSSMETER context for Tecnalía's developers.

OSSMETER technologies will provide significant improvements to industry and have a remarkable overall readiness for industrial use.

During the Eclipse.org use case evaluation Tecnalía has detected different improvements OSSMETER technologies could provide not only on the Eclipse.org context but to current Tecnalía's products and services.

Tecnalía has different tools, services and methodologies to develop and implement Software Product Lines and Software Factories. Usually this approach starts with the static analysis of the current software products / code by a team of Tecnalía's engineers and the client personnel. Using OSSMETER technologies to study different metrics over the code can even help to identify common and variable code inside final products.

With OSSMETER technologies Tecnalía is starting different R&D initiatives about legacy code analysis in the context of software variability management theories.

In the context of software quality assurance and software development improvement Tecnalía's consultancy services are experts in dealing with CMMI for Development, UNE-EN ISO 9001:2000 and GQM ("goal, question, metric", is an approach to software metrics). During the Eclipse.org use case evaluation this team has seen the OSSMETER technologies and they are interested in their possibilities as an assessment tool to pre-evaluate quality concepts. This is a really early idea because the use of the OSSMETER platform as a tool for quality assurance support is completely out of the current project but will be developed by Tecnalía.

Finally Tecnalía is software assessor and technology consultant for the Basque Government in Spain. Software and people certification is a current line of work

between both organizations. From Tecnia's perspective using OSSMETER technologies to generate metrics and objective (and aseptic) comparisons between different public tool providers solutions could be a really interesting service that not only the Basque Government but any public organization can demand in the future.

Tecnia is currently studying the inclusion of OSSMETER ideas and technologies in its future offer to the Basque Government.